CLAIMS

1. A method for establishing a location transparent event handler comprising the steps of:

establishing a Notifier object in a client application for execution in a first process address space, said Notifier object based upon a Notifer class, said Notifier object having a list of Listener objects to be notified upon an event occurrence;

establishing a Listener object in a server application for execution in a second process address space separate from said first process address space, said Listener object based upon a Listener class, said Listener object defining a method to be called upon the occurrence of said event, said Listener object enabled to be callable from said Notifier object; and,

generating a Listener object stub for said Listener object, said Listener object stub configured to be added to said list of Listener objects in said Notifier object, said Listener object stub further configured to remotely call said defined method in said Listener object in response to receiving notification of an event from said Notifier object,

whereby upon said event occurrence, said Notifier object can traverse said list of Listener objects and can notify said Listener object stub of said event occurrence thereby creating a remote call to said defined method in said Listener object.

2. The method of claim 1, wherein said Notifier and Listener classes are Java classes and said first and second process address spaces are in first and second Java Virtual Machines, respectively.

3. The method of claim 2, wherein said generating step comprises the steps of:

RMI compiling said Listener class, said RMI compilation generating said Listener object stub; and,

registering said Listener object with an RMI Registry, said RMI Registry executing in a third Java Virtual Machine,

said Notifier object retrieving a reference to said registered Listener object from said RMI Registry upon said addition said Listener object stub to said list of Listener objects,

said Listener object stub remotely calling said defined method in said Listener object through said retrieved reference upon receiving notification of an event from said Notifier object.

4. A method for performing location transparent event handling comprising the steps of:

creating an instance of a Notifier class in a first process address space, said Notifier instance having a list of Listener objects to be notified upon an event occurrence;

creating an instance of a Listener class in a second process address space, said Listener instance having a method to be called upon the occurrence of said event, said Listener instance enabled to be callable from said Notifier instance;

inserting a Listener object stub in said list of Listener objects in said Notifier instance in said first process address space, said Listener object stub configured to remotely call said defined method in said Listener instance;

receiving an event occurrence in said Notifier instance; and,

responsive to receiving said event occurrence, traversing said list of Listener objects, passing said event to said Listener object stub, creating in said Listener object stub a remote call to said defined method in said Listener instance, and executing said defined method in said Listener instance.

5. The method of claim 4, wherein said Notifier and Listener classes are Java classes and said first and second process address spaces are in first and second Java Virtual Machines, respectively.

6. The method of claim 5, wherein said Listener object stub is generated in an RMI compilation process.

7. The method of claim 6, wherein said inserting step further comprises the step of:

registering said Listener instance with an RMI Registry, said RMI Registry executing in a third Java Virtual Machine,

said Notifier instance retrieving a reference to said registered Listener instance from said RMI Registry upon inserting said Listener object stub to said list of Listener objects.

8. The method of claim 7, wherein said step of creating in said Listener object stub remotely calls said defined method in said Listener instance through said retrieved reference upon receiving said event from said Notifier instance.

9. A machine readable storage, having stored thereon a computer program having a plurality of code sections for establishing a location transparent event handler, said code sections executable by a machine for causing the machine to perform the steps of:

establishing a Notifier object in a client application for execution in a first process address space, said Notifier object based upon a Notifer class, said Notifier object having a list of Listener objects to be notified upon an event occurrence;

establishing a Listener object in a server application for execution in a second process address space separate from said first process address space, said Listener object based upon a Listener class, said Listener object defining a method to be called upon the occurrence of said event, said Listener object enabled to be callable from said Notifier object; and,

generating a Listener object stub for said Listener object, said Listener object stub configured to be added to said list of Listener objects in said Notifier object, said

Listener object stub further configured to remotely call said defined method in said Listener object in response to receiving notification of an event from said Notifier object,

whereby upon said event occurrence, said Notifier object can traverse said list of Listener objects and can notify said Listener object stub of said event occurrence thereby creating a remote call to said defined method in said Listener object.

10. The machine readable storage of claim 9, wherein said Notifier and Listener classes are Java classes and said first and second process address spaces are in first and second Java Virtual Machines, respectively.

11. The machine readable storage of claim 10, wherein said generating step comprises the steps of:

RMI compiling said Listener class, said RMI compilation generating said Listener object stub; and,

registering said Listener object with an RMI Registry, said RMI Registry executing in a third Java Virtual Machine,

said Notifier object retrieving a reference to said registered Listener object from said RMI Registry upon said addition said Listener object stub to said list of Listener objects,

said Listener object stub remotely calling said defined method in said Listener object through said retrieved reference upon receiving notification of an event from said Notifier object.

12. A machine readable storage, having stored thereon a computer program having a plurality of code sections for performing location transparent event handling, said code sections executable by a machine for causing the machine to perform the steps of:

creating an instance of a Notifier class in a first process address space, said

Notifier instance having a list of Listener objects to be notified upon an event occurrence;

creating an instance of a Listener class in a second process address space, said Listener instance defining a method to be called upon the occurrence of said event, said Listener instance enabled to be callable from said Notifier instance;

inserting a Listener object stub in said list of Listener objects in said Notifier instance in said first process address space, said Listener object stub configured to remotely call said defined method in said Listener instance;

receiving an event occurrence in said Notifier instance; and,

responsive to receiving said event occurrence, traversing said list of Listener objects, passing said event to said Listener object stub, creating in said Listener object stub a remote call to said defined method in said Listener instance, and executing said defined method in said Listener instance.

13. The machine readable storage of claim 12, wherein said Notifier and Listener classes are Java classes and said first and second process address spaces are in first and second Java Virtual Machines, respectively.

14. The machine readable storage of claim 13, wherein said Listener object stub is generated in an RMI compilation process.

15. The machine readable storage of claim 14, wherein said inserting step further comprises the step of:

registering said Listener instance with an RMI Registry, said RMI Registry executing in a third Java Virtual Machine,

said Notifier object retrieving a reference to said registered Listener object from said RMI Registry upon inserting said Listener object stub to said list of Listener objects.

16. The machine readable storage of claim 15, wherein said step of creating in said Listener object stub remotely calls said defined method in said Listener instance through said retrieved reference upon receiving said event from said Notifier instance.